



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A  
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**APLIKACE ALGORITMŮ PREDIKTIVNÍ ÚDRŽBY NA  
MONITOROVÁNÍ STAVU EXPERIMENTÁLNÍHO  
PNEUMATICKÉHO ZAŘÍZENÍ**

APPLICATION OF PREDICTIVE MAINTENANCE ALGORITHMS FOR STATE MONITORING OF AN  
EXPERIMENTAL PNEUMATIC DEVICE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Petr Šťastný**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Barnabás Dobossy**

**BRNO 2021**

# Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky  
Student: **Petr Šťastný**  
Studijní program: Aplikované vědy v inženýrství  
Studijní obor: Mechatronika  
Vedoucí práce: **Ing. Barnabás Dobossy**  
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## **Aplikace algoritmů prediktivní údržby na monitorování stavu experimentálního pneumatického zařízení**

### **Stručná charakteristika problematiky úkolu:**

V době Industry 4.0 je snaha vyvinout zařízení, která jsou vybavena širokou škálou senzorů za účelem monitorování jejich aktuálního stavu a v případě potřeby naplánovat nutnou údržbu. Prediktivní údržba je relativně nová oblast, která se zabývá sledováním stavu komponentů za účelem předcházet jejich poruše.

Tato práce se zabývá implementací algoritmů založených na indikátorech stavu za účelem monitorování stavu pneumatického zařízení a v případě výskytu poruchy identifikovat jejich zdroje. Cílem práce je také stanovit minimální nutný počet senzorů z hlediska spolehlivého fungování algoritmu.

**Cíle bakalářské práce:**

1. V rámci rešeršního studia představte oblast prediktivní údržby, a přitom se soustředte na metody založené na stavových indikátorech.
2. Zapojte a zkalibrujte senzory na zkušebním pneumatickém zařízení.
3. Provedte zadanou sadu experimentů a naměřená data upravte do vhodného formátu pro další zpracování.
3. Využijte nástroj Diagnostic Feature Designer v prostředí Matlab pro zpracování naměřených dat za účelem identifikace poruch.
4. Na základě výsledků z bodu 4. zvolte minimální počet senzorů, které umožňují spolehlivé monitorování stavu zařízení a detekci poruch.

**Seznam doporučené literatury:**

VALÁŠEK, Michael. Mechatronika. Vydavatelství ČVUT, 1995. ISBN 80-01-01276-X.

ISERMANN, Rolf. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance.

1. Springer-Verlag Berlin Heidelberg, 2006. ISBN 978-3-540-24112-6.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## Abstrakt

Tato bakalářská práce se zabývá hledáním indikátorů stavu pneumatického zařízení za použití algoritmů strojového učení a vytěžování dat. Cílem bylo určit měřitelnou veličinu a algoritmus jejího vyhodnocování, pomocí kterého bude možné identifikovat stav a zdroje poruch. Data o chování pneumatického válce byla získávána na testovacím zařízení, které bylo osazeno senzory 16 různých veličin. Zpracování a vyhodnocování dat proběhlo v nástrojích Matlabu, konkrétně Diagnostic Feature Designer a Classification Learner.

## Abstract

This bachelor thesis deals with finding state indicators of pneumatic device using algorithms of machine learning and data mining. The goal was to determine measurable quantity and algorithm of its evaluating, using which would be possible to identify state and sources of failures. The data of behavior of pneumatic cylinder were acquired on testing stand, which was equipped by sensors of 16 different quantities. Postprocessing and evaluating of the data took place in Matlab tools, particularly Diagnostic Feature Designer and Classification Learner.

## Klíčová slova

Prediktivní údržba, Průmysl 4.0, strojové učení, SVM, selekce atributů, atributy dat, vytěžování dat, indikátor stavu, pneumatický válec, Diagnostic Feature Designer, Classification Learner

## Keywords

Predictive maintenance, Industry 4.0, machine learning, SVM, feature selection, data features, data mining, state indicator, pneumatic cylinder, Diagnostic Feature Designer, Classification Learner

## Bibliografická Citace

ŠŤASTNÝ, P. *Aplikace algoritmů prediktivní údržby na monitorování stavu experimentálního pneumatického zařízení*. Brno, 2021. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Vedoucí práce: Ing. Barnabás Dobossy.

Prohlašuji, že jsem tuto bakalářskou práci na téma *Aplikace algoritmů prediktivní údržby na monitorování stavu experimentálního pneumatického zařízení* vypracoval samostatně s použitím zdrojů uvedených v seznamu literatury.

**Petr Šťastný**

Brno . . . . .

. . . . .

Děkuji vedoucímu Ing. Barnabási Dobbosymu za spoustu cenných rad, trpělivost a ochotu při vedení mé bakalářské práce.

**Petr Šťastný**

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Rešerše</b>	<b>10</b>
2.1	Prediktivní údržba v kontextu Industry 4.0 . . . . .	10
2.2	Postup implementace prediktivní údržby . . . . .	11
2.3	Atributy dat . . . . .	13
2.4	Výběr atributů . . . . .	13
2.4.1	Filter . . . . .	14
2.4.2	Wrapper . . . . .	15
2.4.3	Embedded . . . . .	15
2.5	Machine learning . . . . .	15
2.5.1	Discriminant analysis . . . . .	15
2.5.2	Support vector machine . . . . .	16
2.6	Problém přetrénování . . . . .	18
2.7	Nástroje Matlabu . . . . .	19
2.7.1	Diagnostic Feature Designer . . . . .	19
2.7.2	Classification learner . . . . .	20
<b>3</b>	<b>Praktická část</b>	<b>21</b>
3.1	Shrnutí cílů . . . . .	21
3.2	Pneustand . . . . .	21
3.3	Sběr dat . . . . .	23
3.3.1	NI DAQ . . . . .	23
3.3.2	Měřené veličiny . . . . .	24
3.4	Modelové situace . . . . .	24
3.5	Výpočet atributů . . . . .	26
3.6	Zhodnocení atributů . . . . .	26
3.6.1	Postup vyhodnocení naměřených dat . . . . .	26
3.6.2	SVM volba senzoru . . . . .	27
3.6.3	ReliefF ohodnocení . . . . .	29
3.6.4	SVM volba atributů . . . . .	29
3.7	Shrnutí algoritmu . . . . .	31
3.8	Vytipovaný senzor . . . . .	32
<b>4</b>	<b>Závěr</b>	<b>33</b>
	<b>Seznam zkratek</b>	<b>34</b>





# 1 Úvod

Prediktivní údržba je moderním přístupem k údržbě strojů a je považována za jeden z principů Industry 4.0. Ten klade důraz na digitalizaci a automatickou komunikaci jednotlivých prvků výrobního procesu i výrobku samotného, díky tomu se výroba stává pružnější, efektivnější a přináší nové nástroje pro kontrolu a optimalizaci všech prvků jednotlivě i procesů jako celku. Stroje se stávají více a více komplexní a jsou vybaveny čím dál větším počtem akčních prvků i snímačů. S rostoucí komplexností strojů rostou také nároky na jejich údržbu. Ta se stejně jako stroje a technologie vyvíjí a přináší nové možnosti diagnostiky a predikce stavů zařízení. Prediktivní forma údržby řeší spolehlivost tak, že diagnostickými metodami hodnotí aktuální stupeň opotřebení a na základě zkušeností a určitých kritérií stanoví moment, kdy je optimální čas pro servis nebo výměnu.

Výhody, které prediktivní údržba přináší jsou:

- životnost komponent je využita v maximální možné a zároveň bezpečné míře
- lepší předcházení poruchám
- nedochází k neplánovaným odstávkám a ztrátám

Mezi nevýhody nebo lépe řečeno nároky tohoto přístupu patří:

- náklady na monitorování a diagnostiku
- pracovníci údržby nejsou už jen odborníci na mechaniku a elektroniku ale musí mít alespoň základní znalosti z oblasti zpracování informací a algoritmizace

V současné době se o prediktivní údržbě v kruzích managementu hodně mluví, skutečných funkčních aplikací však neexistuje mnoho. Trend posledních let ovšem ukazuje, že si podniky uvědomují přínosy prediktivního přístupu a lze očekávat, že se bude tento přístup do vývoje a výroby propisovat více.

Tato bakalářská práce se věnuje práci s daty naměřenými na experimentálním pneumatickém zařízení osazeného velkým množstvím snímačů různých veličin. Cílem je identifikovat jaké snímače jsou vhodné pro spolehlivou identifikaci stavu a zdrojů poruch tohoto zařízení. Pro identifikaci jsou použity nástroje statistické analýzy signálu a strojového učení.

## 2 Rešerše

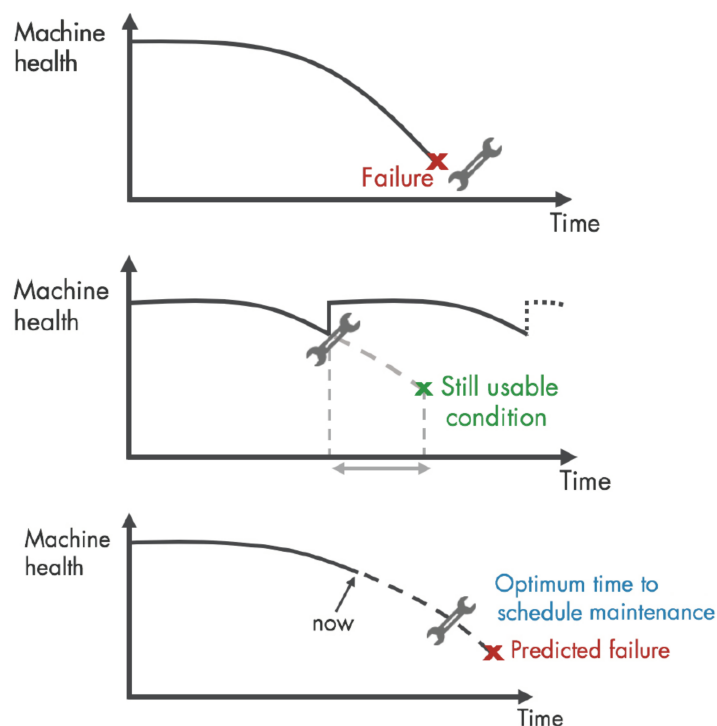
V úvodu této kapitoly popisují přístupy, jak je možné k údržbě strojních zařízení přistupovat. Následně představují proces implementace prediktivní údržby do provozu a všechny jeho etapy. Další dvě podkapitoly popisují získávání a vyhodnocování atributů naměřených dat a metody strojového učení. V poslední podkapitole popisují nástroje prostředí Matlab, které je možné použít pro identifikaci vhodných veličin a jejich atributů a trénování klasifikátoru pomocí strojového učení.

### 2.1 Prediktivní údržba v kontextu Industry 4.0

Informace pro následující kapitolu jsou čerpané z: [1, 14].

Prediktivní údržba je jedním z pilířů probíhající digitalizace průmyslu – Industry 4.0. Podniky se postupně aplikují prvky internetu věcí (IoT), které umožňují průběžné získávání dat o chování sledovaných systémů. Tyto data jsou současně vyhodnocována a poskytují informace, pomocí kterých dokážeme identifikovat aktuální stav a předpovědět kdy dojde k selhání daného prvku. Díky tomu dokážeme zvýšit dobu efektivního používání.

Z pohledu doby okamžiku, kdy je údržba prováděna ji můžeme rozdělit na tři skupiny:



Obrázek 2.1: Srovnání (shora) reaktivní, preventivní a prediktivní údržby. Zdroj: [1]

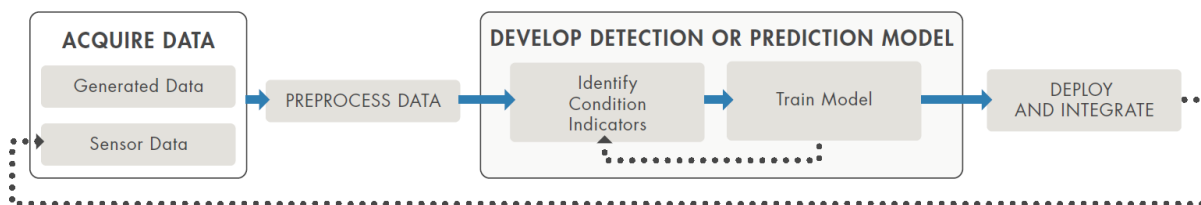
- **Reaktivní údržba** nastává až v okamžiku selhání komponenty. Při provozu nového stroje s nulovou poruchovostí se může zdát, že v tomto období není třeba údržby. To ovšem není pravda, tento laxní přístup k ošetřování stroje se v budoucnu projeví ve snížení životnosti a vyšší udržovací náklady. Reaktivní přístup k údržbě s sebou nese velké riziko neplánovaných odstávek stroje, a tudíž i velkých ztrát.
- **Preventivní údržba** probíhá buď v časově pravidelných intervalech, nebo na základě překročení limitní úrovně sledované veličiny (tento způsob může být označen jako diagnostická údržba). Cílem je provádět údržbu dříve, než nastane porucha. Z důvodu neznalosti bodu, kdy dojde k poruše, musíme být v plánování časů údržby konzervativní. Dle [14]: *Prováděním preventivní údržby lze dosáhnout či prodloužit provozní životnost a spolehlivost zařízení či konstrukčního dílu. Tato skutečnost má samozřejmě i ekonomický dopad. Průzkumy uvádějí průměrnou úsporu nákladů preventivní údržby v rozmezí 12% - 18% v porovnání s čistě reaktivní údržbou.*
- **Prediktivní údržba** je moderním způsobem, jak k údržbě přistupovat. Její podstata spočívá v měření a analýze chování stroje a klade si za cíl včas upozornit na vznikající degradaci nebo neobvyklé chování. Díky tomu, že s určitou přesností víme, v jakém stavu se stroj nachází, dokážeme optimalizovat okamžik údržby. Díky tomu je možné využít na maximum dobu spolehlivého fungování zařízení s vysokou pravděpodobností, že nedojde k nenadálé poruše. Kromě předpovědi selhání si prediktivní údržba klade za cíl identifikaci zdrojů problémů a poruch, díky čemuž víme, co přesně opravit, nebo optimalizovat. Protože jsou odstávky stroje plánované, je možné provést včas potřebnou výrobu a na údržbu se materiálně i informativně připravit. Díky tomu jsou ekonomické následky v porovnání s jinými přístupy k údržbě minimální.

Z tohoto přehledu vyplývá, že prediktivní přístup k údržbě je nejefektivnější ze všech přístupů. V následujících kapitolách se věnuji popisu procesu implementace prediktivní údržby a několika nástrojům, které je možné použít pro analýzu naměřených dat.

## 2.2 Postup implementace prediktivní údržby

Následující kapitola je inspirována: [1, 3].

Postup implementace prediktivní údržby se bude pro různé aplikace lišit. Můžeme jakousi obecnou strukturu, kterou se proces bude řídit



Obrázek 2.2: Schéma kroků při nasazování prediktivní údržby. Zdroj: [2]

- **Acquire data:** Vývoj začíná sběrem velké sady dat, která popisuje sledovanou soustavu ve správných i chybných stavech. Data získávaná za různých provozních podmínek, nám pomůžou k vytvoření robustnějšího a více obecného algoritmu, který dokáže stavy lépe identifikovat. Pro získání dat je možné použít dva přístupy:

- **Data based** je tradiční přístup, kdy data získáváme měřením na reálném zařízení. Tento přístup je časově náročný a hrozí riziko defektu při zatěžování nestandardními okolními vlivy.
- **Model based** je přístup použitelný v situaci, kdy známe soustavu natolik dobře, že dokážeme vytvořit dostatečně přesné digitální dvojče. Na něm jsou simulovány různé okolní podmínky a stavy a je pozorováno její chování. To přináší obrovskou výhodu urychlení testování při vývoji, protože zařízení není fyzicky třeba. Další výhodou této metody je, že nehrozí poškození zařízení při měření, jako tomu je v Data based přístupu. Pokud ovšem soustavu neznáme dostatečně dobře, výsledky budou zkreslené a nebudou odpovídat reálnému chování.
- **Preprocess data:** Dalším krokem je předzpracování získaných dat. Úkolem tohoto kroku je upravit data tak, aby z nich šlo následně identifikovat indikátory stavu, tzv. příznaky. Do zpracování lze zahrnout:
  - Odstranění nerelevantních a chybějících hodnot, odstranění offsetu, detrendace.
  - Redukce šumu, jeho filtrování nebo vyhlazování.
  - Transformace mezi časovou a frekvenční doménou.
  - Pokročilé zpracování signálu jako např. krátkodobá Fourierova transformace.

Z takto upravených dat můžou být statistickou analýzou vytvořeny kvantitativní veličiny charakterizující naměřená data (tzv. atributy).

- **Develop detection or prediction model:** Následuje proces identifikace vhodných příznaků. Existují tři přístupy:
  - **Filter** přístup na základě používá pokročilých statistických nástrojů pro identifikaci těch příznaků, které se předvídatelně mění s různými stavy. Tyto příznaky jsou následně využity pro rozlišování mezi stavy. Jeho výhodou je nízká výpočetní náročnost v porovnání s model based přístupem, zvláště pokud je potřeba vybrat z velkého množství příznaků.
  - **Wrapper a Embedded** jsou dvě metody přistupující k výběru příznaků stavu jinak. Příznaky jsou vybírány na základě přesnosti modelu, který vznikl tréninkem na těchto příznacích. Tento přístup je sice výpočetně náročnější, jeho výsledky jsou však obecnější, protože výběr probíhá na základě společného chování vybraných příznaků.

Trénováním vyhodnocovacího modelu metodami strojového učení může vést k různým funkcím (detekce chybových stavů, detekce anomálií, nebo predikování zbývajících životnosti stroje) podle toho, jakým způsobem budeme předkládat tréninková data. Je potřeba myslet na to, že data je nutné rozdělit na tréninková a testovací, z důvodu ověření spolehlivého fungování navrhnutého algoritmu na nových datech.

- **Deploy and integrate:** Finální fází je aplikování a testování modelu na reálném systému. Model může být implementován přímo na pozorované zařízení nebo v nějakém cloudovém řešení. Po nasazení nastává období testování, kdy je potřeba ověřit správné fungování modelu, případně ladit parametry modelu nebo celý proces vývoje iterovat, dokud nedostaneme kýžené výsledky.

## 2.3 Atributy dat

Pro algoritmy vybírání vhodných příznaků a trénování vyhodnocovacího modelu je potřeba naměřená data dokázat kvantitativně charakterizovat. Pro tento účel dobře slouží atributy popisující signál v časové nebo frekvenční oblasti. Ty můžou mít, pro na první pohled nerozeznatelné signály, velkou rozlišovací schopnost.

Ovšem ne všechna data mají pro rozhodování nějaký informační přínos. Kvalita jejich výběru ovlivňuje kvalitu výsledného rozhodovacího modelu. Díky tomu, že data dokážeme pomocí atributů kvantifikovat, je snadnější je algoritmicky ohodnotit a vybrat mezi nimi ty, které budou pro vyhodnocování nejvhodnější.

Kromě základních statistických veličin jako průměr (Mean), kvadratický průměr (RMS), směrodatná odchylka (Std), nabízí Diagnostic feature designer (DFD – popsán v kapitole 2.7.1) výpočet následujících:

Tabulka 2.1: Další atributy signálu

Feature	Atribut	Co vyjadřuje
<b>Časová doména</b>		
Shape factor	Charakteristika tvaru signálu	Poměr mezi kvadratickým průměrem a průměrem absolutních hodnot
Kurtosis	Koeficient špičatosti	Porovnává rozdělení s normálním rozdělením pravděpodobnosti
Skewness	Koeficient šikmosti	Popisuje nesymetrii rozdělení
Peak value	Výška špičky signálu	Absolutní maximální hodnota signálu
Crest Factor	Činitel výkyvu	Porovnává výšku špičky s kvadratickým průměrem
Impulse Factor	Činitel impulsu	Porovnává výšku špičky s průměrem
Clearance Factor	Činitel vůle	Výška špičky dělená průměrnou výškou špiček
vSigna-to-Noise Ratio	Poměr signálu vůči šumu	Srovnává výkon žádaného signálu vůči výkonu šumu
vTotal Harmonic Distortion	Celkové harmonické zkreslení	Vyjadřuje zkreslení sinusového signálu
Signal to Noise nad Distortion Ratio – SINAD	Poměr signálu vůči šumu + zkreslení	Porovnává výkon signálu se součtem výkonů šumu a zkreslení
<b>Frekvenční doména</b>		
Peak amplitude	Výška špičky	Výška špičky amplitudy ve frekvenční oblasti
Peak frequency	Frekvence špičky	Frekvence špičky ve frekvenční oblasti
Band power	Výkon pásma	Výkon všech harmonických v daném pásmu

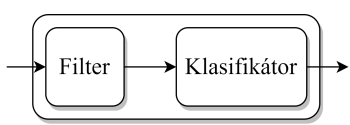
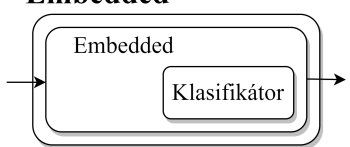
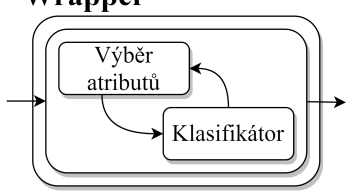
## 2.4 Výběr atributů

Výběr vhodných atributů použitých pro trénovací algoritmus vede ke zlepšením v rychlosti učení, zvýšení srozumitelnosti modelu, i zlepšení chování modelu na obecných datech.

Kromě toho, snížení počtu atributů, vede ke snížení nákladů na získávání dat a lepšímu porozumění měřeným veličinám. [4]

Úkolem algoritmu výběru atributů je určení takových atributů, pomocí kterých dokážeme predikovat nebo identifikovat stav systému.

Algoritmus výběru atributů a trénovací algoritmus jsou úzce svázány. V závislosti na způsobu předávání výsledků mezi těmito algoritmy můžeme mluvit o třech hlavních přístupech: Filter, Embedded a Wrapper.

Metoda	Výhody	Nevýhody
<b>Filter</b> 	<ul style="list-style-type: none"> <li>• Nezávislost klasifikátoru</li> <li>• Nižší výpočetní náklady než Wrapper</li> <li>• Rychlé</li> <li>• Dobrá obecnost</li> </ul>	<ul style="list-style-type: none"> <li>• Žádná interakce s klasifikátorem</li> </ul>
<b>Embedded</b> 	<ul style="list-style-type: none"> <li>• Interakce s klasifikátorem</li> <li>• Nižší výpočetní náklady než Wrapper</li> <li>• Zohledňuje závislost atributů</li> </ul>	<ul style="list-style-type: none"> <li>• Výběr atributů je závislý na klasifikátoru</li> </ul>
<b>Wrapper</b> 	<ul style="list-style-type: none"> <li>• Interakce s klasifikátorem</li> <li>• Zohledňuje závislost atributů</li> </ul>	<ul style="list-style-type: none"> <li>• Výpočetně náročné</li> <li>• Riziko přetrévání</li> <li>• Výběr atributů je závislý na klasifikátoru</li> </ul>

Obrázek 2.3: Srovnání metody výběru atributů. Zdroj: [4]

### 2.4.1 Filter

Filtrační metody hodnotí atributy dle daných pravidel nezávisle na trénovacím algoritmu. Vstupem pro trénovací algoritmus je potom sada nejlepších atributů. Z toho vyplývající výhoda je nízká výpočetní náročnost v porovnání s Embedded nebo Wrapper metodami. Obecně se ale dá říct, že filtrační přístup dává horší výsledky než dvě výše zmíněné metody. [4, 5]

Níže je představen jeden zástupce filtrů.

**ReliefF** je rozšířením algoritmu Relief. Ten pracuje tak, že postupně náhodně vybírá vzorky dat a zjišťuje vzdálenost k nejbližšímu prvku stejné třídy a nejbližšímu z třídy opačné. Následně jsou normalizované vzdálenosti k nejbližším prvkům stejné třídy odečteny od váhy a vzdálenosti k nejbližším prvkům opačné třídy k váze přičteny. Důsledkem toho, že zjišťované vzdálenosti jsou poděleny počtem testovaných prvků, je zajištěno, že velikost výsledné váhy leží v intervalu  $\langle -1; 1 \rangle$ . Čím vhodnější atribut je, tím budou vzdálenosti k nejbližším sousedům stejné třídy menší než vzdálenosti k nejbližším sousedům opačné třídy a váha se bude blížit k 1. Nástavba této metody ReliefF přináší možnost pracovat s daty z více tříd najednou a zároveň je přesnější při práci s nekompletními nebo zašuměnými daty. [4, 6]

V [4] provedli komplexní testování celkem 13 Filter, Wrapper i Embedded přístupů výběru atributů. Simulovány byly data korelující samy se sebou, redundantní data, ne-

linearity, šum a nízký počet dat. Metoda ReliefF vykazovala velmi dobré chování pro všechny tyto problémy, proto bude dále v práci využita právě tato metoda.

### 2.4.2 Wrapper

Wrapper přístup vybírá vhodné atributy na základě výsledné přesnosti natrénovaného klasifikátoru. Algoritmus začíná s podmnožinou atributů a pro tyto atributy zkouší natrénovat klasifikační model. Následně mění množinu atributů a iteračně trénuje klasifikátor znova. Při tom je sledována změna přesnosti modelu, způsobená přidáním nebo odebráním atributu. Díky tomu, že je testována použitelnost množiny vybraných atributů, je dosaženo robustnějšího a obecnějšího výsledku než u Filter přístupu.

### 2.4.3 Embedded

Embedded metody se vyznačují tím, že spojují výhody Filter i Wrapper přístupů. Výběr vhodných atributů probíhá uvnitř trénovacího algoritmu. Algoritmus natrénuje model pro všechny atributy a následně zkouší postupně nulovat vliv jednotlivých atributů a pozoruje vliv na výslednou přesnost. Tímto způsobem algoritmus dokáže identifikovat, které atributy jsou pro fungování modelu důležité.

## 2.5 Machine learning

Pro proces implementace prediktivní údržby je vhodné použít nástrojů umělé inteligence. Jejich nasazení nám pomůže získat lepší představu o chování pozorovaného systému. Další výhodou, kterou přináší je identifikace vhodnějších příznaků. Je možné použít dvou přístupů: strojového učení nebo hlubokého učení.

- **Machine learning** (ML) je ta část umělé inteligence, která se stará o tvorbu algoritmů, které na základě nových vstupních dat dokážou predikovat závěry dané situace. Vstupním materiálem pro strojové učení jsou data, u kterých jsou nám známy podmínky a stavy, při kterých byla měřena.
- **Deep learning** je přístup ML, který pro vyhodnocování používá vícevrstevných neuronových sítí. Identifikace parametrů těchto sítí je výpočetně velmi náročná a pro přesné modely je potřeba velká trénovací datová sada. Odměnou za tuto náročnost je však velmi dobrá přesnost a robustnost modelu.

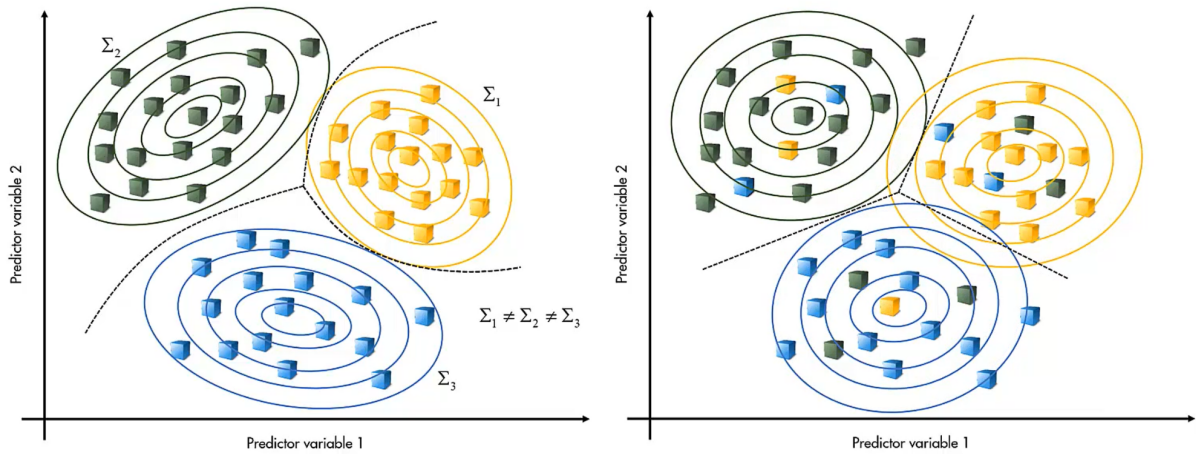
V následující podkapitolách představím dvě metody ML a dále problém přetrénování a jeho řešení.

### 2.5.1 Discriminant analysis

Obsah následující podkapitoly je inspirován [7].

Diskriminační analýza (DA) stojí na předpokladu Gaussova normálního rozdělení prvků jedné třídy. Data jsou však často vícerozměrná. Pro charakterizování vícerozměrného normálního rozdělení použijeme vektor středních hodnot  $\mu$  a kovariační matici  $\Sigma$ , které určují polohu středu a tvar rozdělení. Ze znalosti těchto dvou parametrů můžeme určit křivku, na které je pravděpodobnost pro obě třídy shodná. Takovou křivku je vhodné použít pro hranici dělicí třídy.

Tuto metodu můžeme zjednodušit, pokud předpokládáme standardizované a pro všechny třídy stejné rozložení. Díky tomu budou křivky dělicí prostor přímkami. Toto zjednodušení



Obrázek 2.4: a) Nelineární hranice DA. b) Odolnost DA vůči šumu. Barevné křivky ukazují vrstevnice pravděpodobnosti rozdělení pro danou třídu. Hranice mezi třídami jsou černě tečkované. Zdroj: [7]

se nazývá Linear discriminant analysis a ilustrují ho stejné tvary pravděpodobnostního rozložení jako v obrázku 2.4b.

Důsledkem toho, že metoda pracuje s třídami jako celkem, a ne s jednotlivými měřeními, je robustní vůči šumu, či irrelevantním hodnotám, viz. obrázek 2.4b. Pro rozhodování mezi více než dvěma třídami je hledána rozhodovací křivka pro každou dvojici zvlášť.

### 2.5.2 Support vector machine

Obsah této podkapitoly je inspirován [8, 9].

Support vector machine (SVM) spadá do kategorie jádrových (kernel) algoritmů. Její podstatou je lineární dělení prostoru nadrovinou daného prostoru. Pro 2D je nadrovinou přímka, pro 3D rovina.

V nejjednodušším případě, kdy jsou data lineárně separovatelná lze napsat podmínku:

Předpokládáme, že pro všechny body platí:

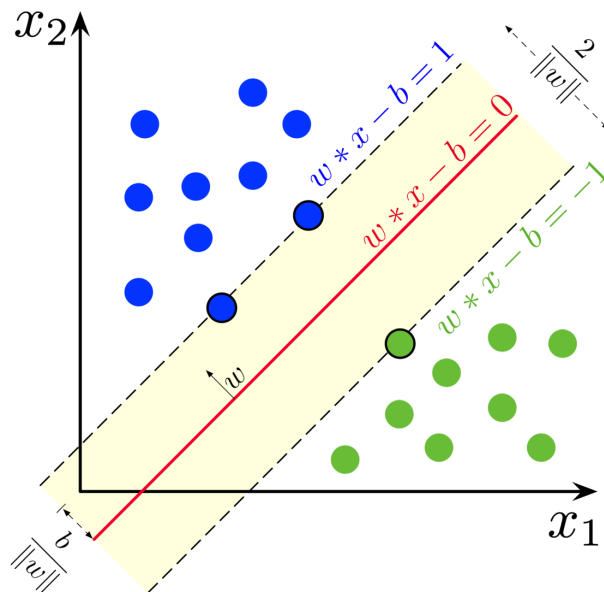
$$\begin{aligned}\vec{x}_i \cdot \vec{w} + b &\geq +1 \\ \vec{x}_i \cdot \vec{w} + b &\leq -1\end{aligned}\tag{2.1}$$

což lze sloučit do jediné nerovnice:

$$y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0 \quad \forall i\tag{2.2}$$

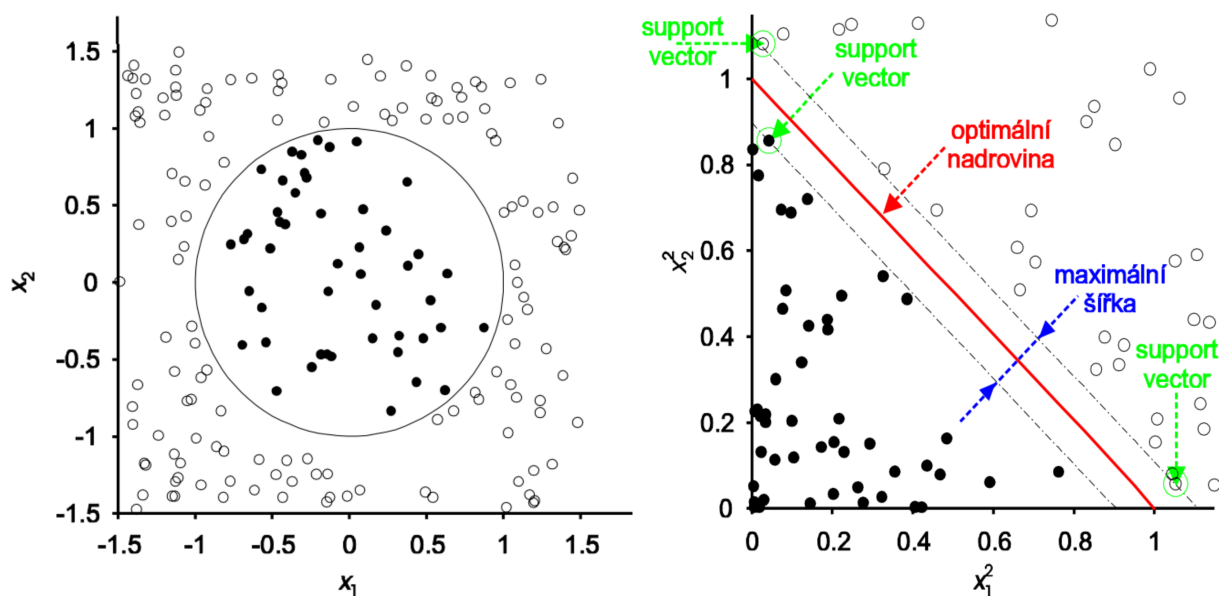
Ta platí pro všechny body  $\vec{x}_i$ . Příslušící jedné ze tříd  $y_i = \{-1; 1\}$ .  $\vec{w}$  je normálový vektor k rozdělovací rovině, dvojnásobek převrácené hodnoty jeho absolutní velikosti  $\frac{2}{|\vec{w}|}$  je šířka pásma a  $\frac{b}{|\vec{w}|}$  je vzdálenost roviny od počátku souřadnic. Algoritmus následně hledá takovou rovinu, která maximalizuje šířku pásma mezi body jednotlivých tříd. Body, pro které má výraz 2.1 hodnotu +1 nebo -1, se nazývají podpůrné vektory. Tyto body tvoří podporu pro oddělovací rovinu a jejich odebrání by změnilo pozici oddělovací roviny.





Obrázek 2.5: Optimální dělicí přímka determinovaná metodou SVM. Zdroj: [8]

Může však nastat situace, kdy lineární dělení prostoru není možné. Využít lze převodu prostoru do jiného, kde lineární rozdělení tříd možné je. Tato transformace je většinou uskutečněna přidáním dalších dimenzí. Obecně můžeme říct, že pro každý prostor s  $N$  prvky existuje lineárně dělitelný prostor s  $N - 1$  nebo více dimenzemi. Transformaci do nového prostoru představuje obrázek 2.6, kde v původním prostoru  $\langle x_1, x_2 \rangle$  nešlo lineárně oddělit. Po transformaci do nového prostoru  $\langle x_1^2, x_2^2 \rangle$  lineární dělení již možné je.



Obrázek 2.6: Transformace do jiného prostoru. Zdroj: [9]

U zašuměných dat nastává problém s identifikací dělicí roviny. Pro řešení tohoto problému může být použita technika soft margin (měkký okraj), která při trénování dovoluje algoritmu udělat určité množství chyb. Algoritmus se potom snaží minimalizovat funkci:

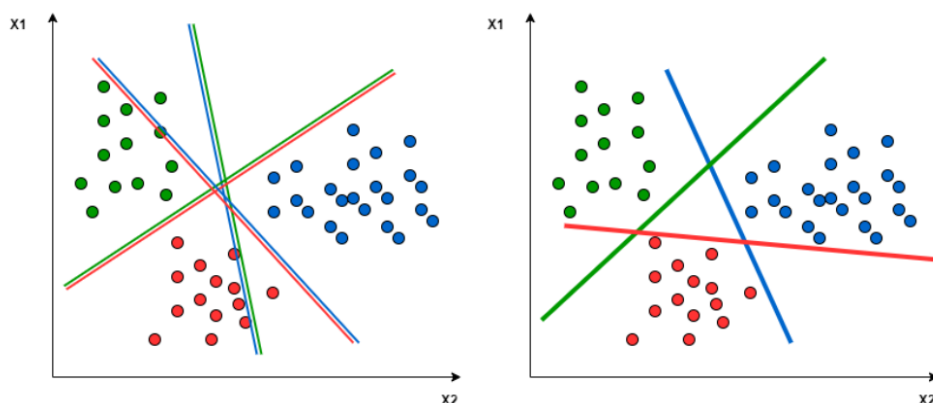
$$L = \frac{1}{2}||w||^2 + C(\text{počet chyby}) \quad (2.3)$$

$||w||$  je velikost normálového vektoru a  $C$  je parametr citlivosti na chyby. Pokud je  $C$  nízké, počet chyb je méně významný a výsledná šířka pásma bude větší než při velkém  $C$ . Optimalizace algoritmu můžeme dosáhnout nastavením vah pro chyby v závislosti na vzdálenosti od dělicí roviny. Prvky vzdálenější od dělicí roviny by měly mít větší penalizaci než prvky blízké.

Základní SVM podporuje klasifikaci pouze mezi dvěma třídami. Pro klasifikaci více tříd je potřeba problém rozdělit na několik binárních klasifikačních problémů. Existují dva přístupy, které tento problém řeší.

One-to-One (jeden na jednoho) hledá nadrovinu každou dvojici tříd zvlášť. Výsledný počet nadrovin je tedy  $\frac{m(m-1)}{2}$ , kde  $m$  je počet tříd.

Druhou možností je One-to-Rest (jeden proti všem), kdy je postupně pro každou třídu hledána nadrovina dělicí třídu od zbytku dat. Počet nadrovin je roven počtu tříd  $m$ .



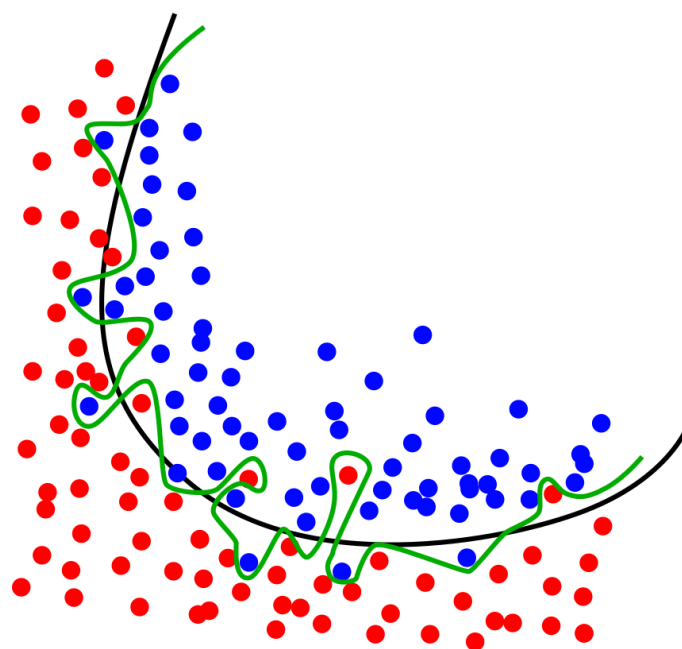
Obrázek 2.7: Hledání optimální dělicí roviny u zašuměných dat. Zdroj: [11]

Pro vyšší počet tříd je metoda One-to-One výrazně výpočetně náročnější, protože hledáme vyšší počet nadrovin než u One-to-Rest přístupu. Odměnou za to je však lepší umístění nadrovin. viz obrázek 2.7.

## 2.6 Problém přetrénování

U klasifikační úlohy ML může nastat, že učící algoritmus sice minimalizuje chybu na trénovacích datech ale na reálných datech model dosahuje velmi nízké přesnosti. Tento stav ilustruje zelená rozhodovací hranice v obrázku 2.8. Vidíme však, že skutečné hranice dělicí třídy se spíš blíží černá čára.

Pro potlačení tohoto problému je možné použít tzv. křížové ověřování (cross validation), která optimalizuje parametry modelu tím způsobem, že data náhodně rozdělí na  $k$  skupin (většinou stačí 5). A následně provede  $k$  trénování modelu tak, že do trénovací sady vybere všechny skupiny až na jednu, na které je testována přesnost modelu. Parametry výsledného modelu jsou na základě přesnosti jednotlivých modelů optimalizovány tak, že výsledek vykazuje obecnější chování a na nových datech bude mít lepší výsledky. [12]



Obrázek 2.8: Hledání optimální dělicí roviny u zašuměných dat. Zdroj: [12]

## 2.7 Nástroje Matlabu

Pro práci s daty za účelem jejich analyzování ve smyslu prediktivní údržby, nabízí Matlab balíček nástrojů Predictive Maintenance Toolbox. Jeho součástí jsou nástroje pro zpracování surových dat, identifikaci příznaků, jejich vyhodnocování, modelování klasifikačních algoritmů a modelování fyzických systémů s možností simulovat poruchové stavy. [13]

### 2.7.1 Diagnostic Feature Designer

Aplikace Diagnostic Feature Designer (DFD) značně usnadňuje a urychluje práci s datovými soubory vytvořenými při měřeních nebo simulacích soustavy. Aplikace v přehledném grafickém prostředí a bez nutnosti psaní kódu umožňuje:

- vizualizace importovaných dat a jejich seskupování podle podmínkových proměnných.
- převod signálu z časové do frekvenční oblasti.
- extrahovat nové atributy popisující data, vizualizovat jejich účinnost pomocí histogramů a ohodnotit je podle několika statistických kritérií.
- počítat atributy z oblastí statistiky signálů, nelineární analýzy, spektrální analýzy a atributy pro rotační stroje.
- všechny vypočítaná data exportovat zpět do Matlabu.
- pro účely automatizace exportovat skript, který na nových dat provede příslušné operace.

### 2.7.2 Classification learner

Aplikace Classification Learner usnadňuje a zpřehledňuje práci při trénování modelů pro klasifikaci dat. Pomocí této aplikace je možné navrhnout algoritmus strojového učení. Uživatel má pod kontrolou výběr použitých atributů, schéma ověřování správnosti modelu i nastavení algoritmu trénování.

Trénovat je možné metodami:

- Decision trees
- Discriminant analysis
- Support vector machines
- Logistic regression
- Nearest neighbours
- Naive Bayes ensemble
- Neural network classification.

V grafickém rozhraní je možné různými metrikami porovnávat přesnost natrénovaných modelů a vyhodnocovat nejlepší z nich.

Zpět do Matlabu je možné exportovat klasifikační model nebo trénovací funkci, která na nových datech natrénuje klasifikační model podle daných parametrů.

Pro přesnější klasifikaci je možné použít nástroj, který pro danou metodu automaticky naladí tzv. hyperparametry modelu. Ty jsou pro různá schémata tréninku různá, např. pro SVM vybere typ kernelu a jeho měřítko, vybere z přístupu One-to-One nebo One-to-All a rozhodne, jestli atributy před klasifikací normalizuje, nebo je ponechá v původním měřítku.

## 3 Praktická část

### 3.1 Shrnutí cílů

Na základě zadání byly stanoveny následující cíle a úkoly:

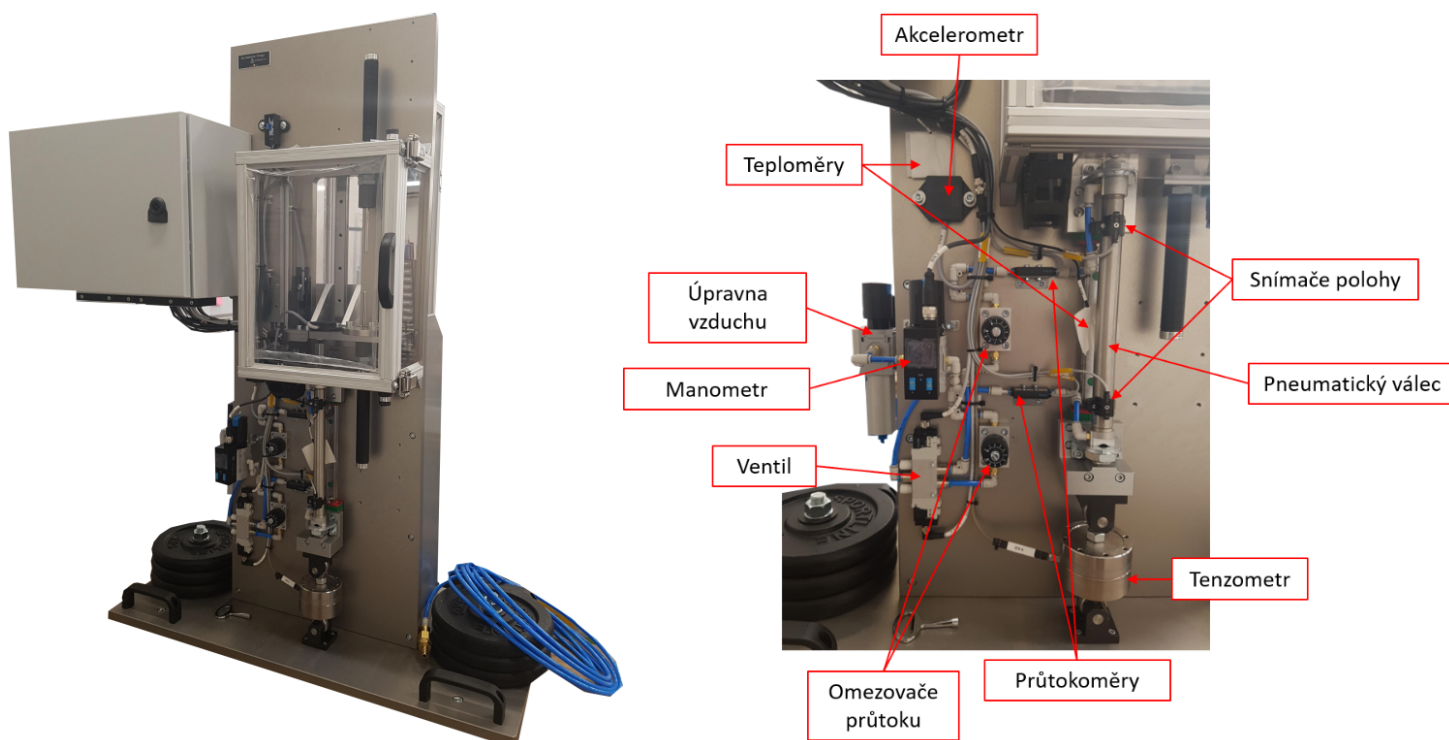
- Zapojení senzorů a elektroniky
- Kalibrace senzorů a zprovoznění sběru dat
- Naprogramování aplikace pro automatizovaný sběr dat, jejich převedení do základních jednotek a ukládání
- Provedení rozsáhlého souboru experimentů s různými nastaveními
- Vyhodnocení dat pomocí aplikace Diagnostic feature designer, Classification learner a vlastních skriptů
- Na základě předchozího bodu vybrat senzor a stanovit algoritmus vhodné pro identifikaci stavů

### 3.2 Pneustand

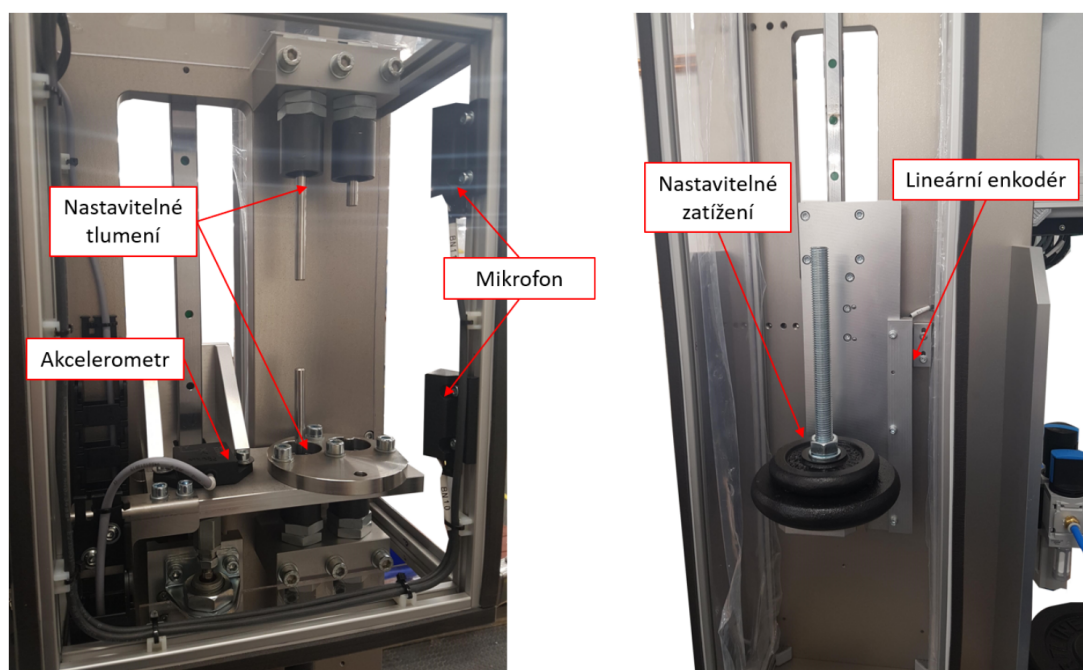
Pneustand je experimentální zařízení osazené jedním pneumatickým válcem, s možností regulace: průtoku vzduchu do válce, tlumení dojezdu do koncových poloh, provozního tlaku a změny zátěže za účelem simulovat chybové nastavení pneumatického zařízení. Zařízení vzniklo ve spolupráci s brněnskou firmou Mechatronic Design & Solutions s.r.o. jako součást programu TRIO, který zajišťuje Ministerstvo průmyslu a obchodu, a který si klade za cíl podpořit průmyslový výzkum a experimentální vývoj.

Pneustand byl při experimentech připojen na rozvodnou síť stlačeného vzduchu v rámci laboratoře. Úpravna vzduch vyfiltruje a redukuje tlak na 6.5 bar. Vzduch následně prochází přes přesný omezovač tlaku s elektronickým manometrem, z kterého byl snímán tlak. Tok vzduchu do válce je přepínán bistabilním elektromagnetickým ventilem. Pro důvody experimentu byly pro regulaci průtoku použity přesné omezovače průtoku, které jsou ve směru toku do válce otevřeny a ve směru z válce ven omezují průtok. Ten je ve směru do válce měřen dvojicí průtokoměrů. Válec je bez vnitřního tlumení s jmenovitým zdvihem 200 mm a vnitřním průměrem 20 mm. Přítomnost v koncových polohách je snímána indukčními snímači.

Válec je v podélné ose pohyblivě uložen a jeho poloha je vymezena tenzometrem. Válec pohybuje s výtahem, který má hmotnost 7 kg. Celkovou hmotnost je možné nakonfigurovat přidáním dalšího závaží. Je možné vybrat ze tří, resp. čtyř kombinací tlumení: pouze dlouhý tlumič – 80 mm, pouze nastavitelný krátký tlumič 20 mm, krátký i dlouhý tlumič, žádný tlumič. Horní i dolní krátký tlumič jsou nastavitelné.

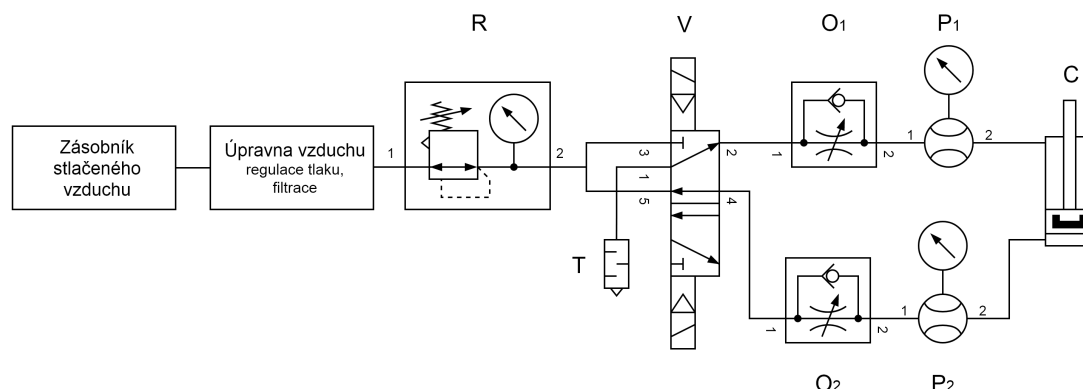


Obrázek 3.1: Pneustand celek a detail přední stana.



Obrázek 3.2: Pnestand detail prostoru výtahu a detail zadní strana.

Poloha soustavy píst-výtah vůči základnímu rámu je měřena lineárním enkodérem její zrychlení je snímáno akcelerometrem, který je umístěn na výtahu. Druhý akcelerometr je upevněn přímo na základně. Síly, kterými působí válec v podélném směru jsou měřeny tenzometrem. Stand je dále vybaven trojicí mikrofونů, kdy dva snímají prostor dolního a horního dorazu a jeden snímá zvuk okolí.

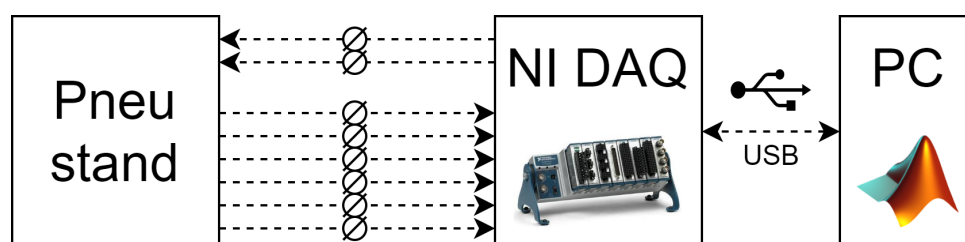


Obrázek 3.3: Schéma pneumatického zapojení.

### 3.3 Sběr dat

#### 3.3.1 NI DAQ

Pro měření a všech signálů byl použit modulární systém CompactDAQ od společnosti National Instruments. Nejprve bylo nutné nainstalovat ovladače aby bylo možné komunikovat přes sběrnici USB s PC. Následně po doinstalování Data Acquisition Toolboxu do Matlabu bylo možné s DAQem komunikovat a konfigurovat jej přímo z Matlabu.



Obrázek 3.4: Schéma sběru dat.

Vlastností tohoto modulárního systému je, že výměnou vstupních a výstupních karet je možné jej uzpůsobit pro potřeby aktuálního měření. Pro sběr signálů byly použity tyto karty:

Tabulka 3.1: Použité karty a měřené veličiny

Označení karty	Veličina rozlišení	Senzory	Vzorkovací frekvence
NI 9221	Napětí $\pm 60V$ , 12 bitů	Průtokoměr 1 a 2, manometr, akcelerometr 1 a 2, tenzometr	1 kHz
NI 9474	Napětí $\pm 60V$ , 12 bitů	Mikrofony: okolí, spodní tlumič, horní tlumič	40 kHz
NI 9411	Digitální 1/0	Snímače koncové polohy: základní a pracovní pozice	1 kHz
NI 9401	Počet tiků enkodéru	Lineární snímač pozice	1 kHz

Pro mikrofony byla zvolena vzorkovací frekvence 40 kHz, kvůli požadavku na násled-

nou transformaci do frekvenční domény, ostatní veličiny byly měřeny s rozlišením 1kHz. Protože dvě různé vzorkovací frekvence není možné spojit do jedné instance měření, bylo nutné vytvořit instance dvě, jednu pro všechny karty měřící na 1 kHz a druhou pro kartu měřící na 40 kHz. Při startu měření bylo nutné obě instance softwarově synchronizovat a měření spustit v jeden okamžik.

### 3.3.2 Měření veličiny

Všechny signály byly na základě charakteristik senzorů přepočítány do základních fyzikálních jednotek.

Tabulka 3.2: Přehled použitých senzorů

veličina	senzor - typ	veličina rozsah	výstup
koncová poloha základní pozice	SMT-8M-A-PS-24V-E-0.3-M8D	nepřítomen přítomen	0/24V
koncová poloha pracovní pozice	SMT-8M-A-PS-24V-E-0.3-M8D	nepřítomen přítomen	0/24V
průtokoměr – vysouvání válce	SFAB-50U-WQ6-2SV-M12	0..50 l/min	0..10V
průtokoměr – zasouvání válce	SFAB-50U-WQ6-2SV-M12	0..50 l/min	0..10V
manometr na redukčním ventilu	MS4-LR-1/8-D7-AD3-AS-WBM	0..10 bar	0..24V
tenzometr	8524-6020-S000S500	-2..+2kN	0..10V
akcelerometr na výtahu	824-4030-006-120	-6..+6g	0,5..4,5V
akcelerometr na základně	824-4030-006-121	-6..+6g	0,5..4,5V
odměřování polohy výtahu	RLS LA11 SQ C 08B B A 10D F 00	0..260mm	tiky enkodéru
mikrofony spodní tlumiče	1612740 - VMA309	48-66dB	0..5V
mikrofony horní tlumiče	1612740 - VMA310	48-66dB	0..5V
mikrofon okolí	1612740 - VMA311	48-66dB	0..5V

## 3.4 Modelové situace

Pro účely demonstrování několika základních aplikačních nasazení pneumatických válců byly specifikovány tři modelové situace:

1. Přesun stanice – různé zatížení, pohyb je tlumen v posledních 20 mm
  - Válec pohybuje s těžkou stanicí
  - Pohyb bez nadměrných rázů
  - Tlumení pouze krátkými tlumiči



## 2. Lisování směrem nahoru

- Typicky situace, kdy ocelová osička (průměr 3 mm) je lisována do plastového housingu
- Tlumení v délce 80 mm před koncovou polohou a posledních 20 mm tlumení ještě dalším tlumičem
- Kontrolovaný pohyb směrem nahoru

## 3. Šroubování / lisování směrem dolů

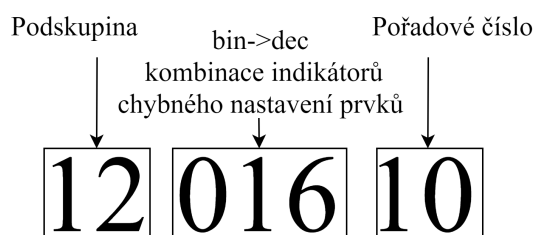
- Nastavení obdobné jako v situaci 2, ovšem zaměřujeme se na kontrolovaný pohyb směrem dolů

Tabulka 3.3: Přehled podkategorií modelových situací - správná nastavení

	Přesun stanice			Lisování nahoru		Lisování dolů	
Označení podskupiny	11xxxxx	12xxxxx	13xxxxx	21xxxxx	22xxxxx	31xxxxx	32xxxxx
Tlak [bar]	6			6		5.5	
Přídavné závaží [kg]	6.25	5	0	0	1.25	0	1.25
Škrticí ventil 1	4	4	3	3	3	5	5
Škrticí ventil 2	2	2	3	3	3	3	3
Tlumič malý spodní	6	6	4	1	1	6	6
Tlumič velký spodní	ne	ne	ne	ano	ano	ano	ano
Tlumič malý horní	4	4	3	4	4	1	1
Tlumič velký horní	ne	ne	ne	ano	ano	ano	ano

Pro každou podkategorie byla naměřena data pro správné nastavení viz. tabulka 3.3 a následně byly postupně měněny parametry omezovačů průtoku a síly tlumení malých tlumičů. Takže pro každou podkategorii vzniklo 35 kombinací nastavení.

Pro zachování přehlednosti byl stanoven systém pro označování tříd. Stav je označen sedmimístným číslem, první dvě číslice označují podskupinu viz. tabulka 3.3 ze které konkrétní stav vychází. Další tři číslice označují do desítkové soustavy převedený binární kód, jehož každý bit odpovídá jednomu nastavitelnému prvku a nabývá hodnoty 1 pokud je prvek nastaven chybně. Poslední dvě číslice jsou pořadové číslo stavu dané podskupiny.



Obrázek 3.5: Struktura značení tříd.

Všechna měření proběhla v 10sekundových cyklech. V první sekundě válec čeká v

základní pozici (dolní), následuje 5sekundový interval v horní pracovní pozici (horní) a cyklus je ukončen zase v základní pozici.

Při prvotním zpracování se ukázalo, že po 10 cyklech známe s dostatečnou přesností, protože jsme ale chtěli robustnější výsledky a případnou možnost rozdělit data na trénovací a testovací, bylo měřeno 20 cyklů pro každý stav.

Data byla měřená pro všechny podkategorie, pro účel zjišťování indikátorů stavu jsem pracoval s daty z podskupin 11xxxxx a 21xxxxx.

## 3.5 Výpočet atributů

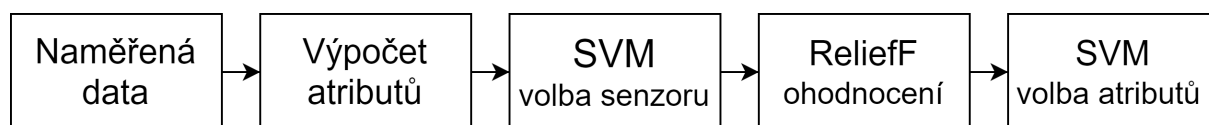
Pro tvorbu atributů z naměřených dat jsem použil nástroj Diagnostic feature designer představený v kapitole 2.7.1.

Pro data z všech senzorů byly identifikovány atributy charakterizující signál v časové doméně, kromě toho pro signály z mikrofónů bylo vypočítáno výkonové spektrum a z něj byly počítány další atributy charakterizující signál ve frekvenční oblasti. Všechny počítané atributy jsou představeny v kapitole 2.3.

## 3.6 Zhodnocení atributů

### 3.6.1 Postup vyhodnocení naměřených dat

Po výpočtu atributů a roztřizení podle chybových stavů je nutné stanovit postup a nástroje, které budou použity pro hodnocení a výběr vhodných indikátorů stavu.



Obrázek 3.6: Workflow hodnocení atributů.

Cílem určit minimální počet senzorů zaručující spolehlivé fungování navrženého algoritmu, nejlépe jen jeden. Proto jsem vhodnost senzorů měřil tak, že jsem trénoval klasifikátor postupně pro každý senzor, na základě všech atributů daného senzoru a porovnával jsem přesnost klasifikátoru. Abych zajistil obecnost výsledku, tak jsem toto provedl pro všechny modelové situace. Pro trénování klasifikátoru jsem zvolil metodu SVM s gaussovským kernelem. K této volbě jsem došel empiricky, kdy SVM s gaussovským kernelem vykazovala nejvyšší přesnost při trénování pro jednotlivé senzory, při zahrnutí všech atributů daného senzoru.

Na základě porovnání různých přístupů vedoucí k identifikaci vhodných indikátorů stavu představených v [4], jsem se rozhodl pro porovnání jednotlivých atributů použít metodu ReliefF, která vykazovala dobré chování pro velký rozptyl modelových situací.

Poslední fází, když známe senzor, který nejlépe indikuje stav válce a známe ohodnocení informační hodnoty jeho atributů můžeme eliminovat ty atributy, které mají nejnižší hodnotu. A díky tomu zjednodušit klasifikační model. Postup je takový, že začneme s nejlepším atributem a natrénujeme pro něj klasifikátor, postupně budeme atributy přidávat podle toho, jak byly v předchozím bodě ohodnoceny. Budeme pozorovat, jak se bude měnit přesnost modelu. Ve chvíli, kdy se přesnost po přidání atributu nebude měnit, přidávání zastavíme.

### 3.6.2 SVM volba senzoru

Prvním krokem vyhodnocení atributů bylo určení jednoho nebo více senzorů, které budou vhodné pro klasifikaci. K ohodnocení senzorů jsem přistoupil tak, že pro každý senzor jsem natrénoval algoritmus, který pro trénink použil všechny atributy tohoto senzoru. Trénink jsem provedl zvlášť pro každou subkategorii chybových stavů (celkem 4 – dva pro tlumení, dva pro omezení průtoku) aby bylo možné rozlišit jak je senzor schopen rozlišovat konkrétní chybové stavy.

Pro trénink byla použita metoda SVM s medium-gauss kernelem. Křížová validace byla nastavena na 5 přeložení. Trénovací algoritmus byl navržen v aplikaci Classification Learner a pro účely automatizovaného výpočtu pro různé senzory byl exportován. Jako měřítko vhodnosti senzoru byla použita přesnost natrénovaného klasifikátoru. Výsledky tohoto procesu jsou znázorněny v tabulkách 3.4, 3.5 a 3.6.

Z výsledků můžeme pozorovat, že úroveň omezení průtoku obstojně detekovalo vícero senzorů. Ovšem s detekcí úrovně nastavení tlumení měla většina senzorů problém. Jediný senzor, který dokázal spolehlivě detekovat omezení průtoku i nastavení tlumení, napříč většinou modelových situací, je lineární enkodér – LeverPosition.

Tabulka 3.4: Výsledky klasifikace podle jednotlivých senzorů - modelová situace 1 (zvýrazněny nejlepší senzory pro daný stav).

Modelová situace	11				12				13			
Zdroj chybového stavu	průtok 1	průtok 2	tlumení spodní	tlumení horní	průtok 1	průtok 2	tlumení spodní	tlumení horní	průtok 1	průtok 2	tlumení spodní	tlumení horní
Třída	11032	11016	11008	11002	12032	12016	12008	12002	13032	13016	13008	13002
FlowExtrusion	0.111	0.125	0.111	0.167	0.115	0.157	0.119	0.095	0.100	0.125	0.125	0.143
FlowContraction	1.000	1.000	0.806	0.825	1.000	1.000	0.819	0.809	1.000	0.988	0.863	0.750
AirPressure	0.844	0.869	0.456	0.500	0.945	0.886	0.600	0.641	0.770	0.781	0.500	0.564
AccelerometerMoving_axisY	0.122	0.144	0.111	0.175	0.100	0.157	0.131	0.118	0.095	0.969	0.113	0.157
AccelerometerMoving_axisZ	0.967	0.994	0.628	0.783	0.920	0.993	0.675	0.709	0.815	0.988	0.638	0.771
AccelerometerStatic_axisY	0.111	0.125	0.111	0.167	0.100	0.143	0.125	0.091	0.100	0.125	0.125	0.143
AccelerometerStatic_axisZ	0.111	0.125	0.111	0.167	0.100	0.143	0.125	0.091	0.100	0.125	0.125	0.143
StrainGauge	0.956	0.956	0.739	0.833	0.980	0.979	0.856	0.741	0.965	0.963	0.900	0.821
ProximitySensor_bottom	0.006	0.088	0.006	0.700	0.075	0.129	0.075	0.223	0.035	0.381	0.069	0.236
ProximitySensor_upper	0.661	0.394	0.217	0.500	0.785	0.864	0.281	0.759	0.760	0.975	0.413	0.729
LeverPosition	0.994	1.000	0.889	0.958	0.990	0.979	0.906	0.986	0.975	0.994	0.869	0.964
MIC_uBumper	0.917	0.950	0.861	0.767	0.970	0.971	0.819	0.768	0.920	0.956	0.856	0.793
MIC_bBumper	0.872	0.963	0.839	0.783	0.965	0.950	0.831	0.755	0.890	0.981	0.831	0.800
MIC_Ambient	0.767	0.925	0.706	0.775	0.960	0.907	0.775	0.714	0.875	0.906	0.781	0.764

Tabulka 3.5: Výsledky klasifikace podle jednotlivých senzorů - modelová situace 2 (zvýrazněny nejlepší senzory pro daný stav).

Modelová situace	21				22			
Zdroj chybového stavu	průtok 1	průtok 2	tlumení spodní	tlumení horní	průtok 1	průtok 2	tlumení spodní	tlumení horní
Třída	21032	21016	21008	21002	21032	21016	21008	21002
FlowExtrusion	0.091	0.100	0.125	0.125	0.091	0.100	0.125	0.143
FlowContraction	<b>1.000</b>	<b>1.000</b>	0.906	0.894	<b>1.000</b>	0.980	0.894	0.829
AirPressure	0.105	0.945	0.575	0.588	0.827	0.965	0.619	0.507
AccelerometerMoving_axisY	0.091	0.100	0.125	0.125	0.100	0.105	0.119	0.150
AccelerometerMoving_axisZ	0.973	0.870	0.488	0.438	0.900	0.850	0.288	0.293
AccelerometerStatic_axisY	0.091	0.100	0.125	0.125	0.091	0.100	0.125	0.143
AccelerometerStatic_axisZ	0.091	0.100	0.125	0.125	0.091	0.100	0.125	0.143
StrainGauge	0.982	0.990	0.806	0.769	0.964	0.965	0.869	0.807
ProximitySensor_bottom	0.000	0.465	0.081	0.725	0.000	0.980	0.169	0.721
ProximitySensor_upper	0.282	0.170	NaN	NaN	0.241	0.175	NaN	0.357
LeverPosition	<b>1.000</b>	<b>1.000</b>	<b>0.981</b>	<b>0.956</b>	<b>0.991</b>	<b>0.995</b>	<b>0.963</b>	<b>0.964</b>
MIC_uBumper	0.955	0.890	0.694	0.725	0.955	0.905	0.575	0.586
MIC_bBumper	0.955	0.950	0.638	0.713	0.945	0.875	0.663	0.657
MIC_Ambient	0.927	0.895	0.606	0.706	0.568	0.860	0.631	0.736

Tabulka 3.6: Výsledky klasifikace podle jednotlivých senzorů - modelová situace 3 (zvýrazněny nejlepší senzory pro daný stav).

Modelová situace	31				32			
Zdroj chybového stavu	průtok 1	průtok 2	tlumení spodní	tlumení horní	průtok 1	průtok 2	tlumení spodní	tlumení horní
Třída	31032	31016	31008	31002	32032	32016	32008	32002
FlowExtrusion	0.091	0.125	0.143	0.167	0.121	0.100	0.125	0.167
FlowContraction	<b>0.991</b>	<b>0.938</b>	0.729	0.908	0.929	0.960	0.713	0.908
AirPressure	<b>0.277</b>	0.838	0.679	0.800	0.683	0.780	0.556	0.817
AccelerometerMoving_axisY	0.091	0.125	0.143	0.167	0.083	0.100	0.125	0.167
AccelerometerMoving_axisZ	0.791	0.906	0.429	0.275	0.708	0.690	0.413	0.308
AccelerometerStatic_axisY	0.091	0.125	0.143	0.167	0.083	0.100	0.125	0.167
AccelerometerStatic_axisZ	0.091	0.125	0.143	0.167	0.083	0.100	0.125	0.167
StrainGauge	0.982	0.944	0.793	0.875	0.850	0.960	0.825	0.925
ProximitySensor_bottom	0.055	0.825	<b>1.000</b>	0.600	0.038	0.775	0.806	0.675
ProximitySensor_upper	0.405	0.750	0.550	0.600	0.429	0.865	0.463	0.708
LeverPosition	0.982	<b>0.975</b>	<b>0.864</b>	<b>0.983</b>	<b>0.925</b>	<b>0.990</b>	<b>0.906</b>	<b>0.983</b>
MIC_uBumper	0.927	0.944	0.879	0.775	0.917	0.920	0.844	0.758
MIC_bBumper	0.941	0.950	0.879	0.633	<b>0.946</b>	0.915	0.856	0.692
MIC_Ambient	0.895	0.888	0.764	0.742	0.888	0.830	0.825	0.700

### 3.6.3 ReliefF ohodnocení

Dalším krokem bylo ohodnocení atributů dat z lineárního enkodéru. Když jsem viděl, že nejlépe vychází snímání, bylo mi vedoucím práce doporučeno zkusit z polohy vypočítat rychlost a zrychlení a zahrnout i tyto veličiny do klasifikačního algoritmu. Polohu jsem tedy dvakrát derivoval a před každou derivací filtroval klouzavým průměrem. Pro průběhy rychlosti a zrychlení jsem pomocí DFD vypočítal atributy viz. kapitola 2.3

Tyto atributy jsem následně ohodnotil podle metody ReliefF popsané v kapitole 2.4.1.

Tabulka 3.7: Ohodnocení atributů podle metody ReliefF.

	Clearance Factor	Crest Factor	Impulse Factor	Kurtosis	Mean	Peak Value	RMS	SINAD	SNR	Shape Factor	Skewness	Standard deviation	THD
Position	0.0019	0.0050	0.0037	0.0000	0.0215	-0.0001	0.0147	-0.0099	-0.0105	0.0114	0.0138	0.0042	-0.0049
Speed	0.0022	0.0023	0.0035	0.0011	0.0067	0.0125	0.0126	-0.0105	-0.0087	0.0146	0.0081	0.0126	-0.0007
Acceleration	0.0103	0.0087	0.0117	0.0231	-0.0022	0.0112	0.0207	0.0071	0.0083	0.0200	0.0433	0.0207	-0.0158

Tabulka 3.8: 15 nejvhodnějších atributů podle hodnocení ReliefF.

Order	Quantity	Metrics
1	Acceleration	Skewness
2	Acceleration	Kurtosis
3	Position	Mean
4	Acceleration	RMS
5	Acceleration	StD
6	Acceleration	Shape factor
7	Position	RMS
8	Speed	Shape factor
9	Position	Skewness
10	Speed	RMS
11	Speed	StD
12	Speed	Peak Value
13	Acceleration	ImpulseFactor
14	Position	Shape factor
15	Acceleration	Peak Value

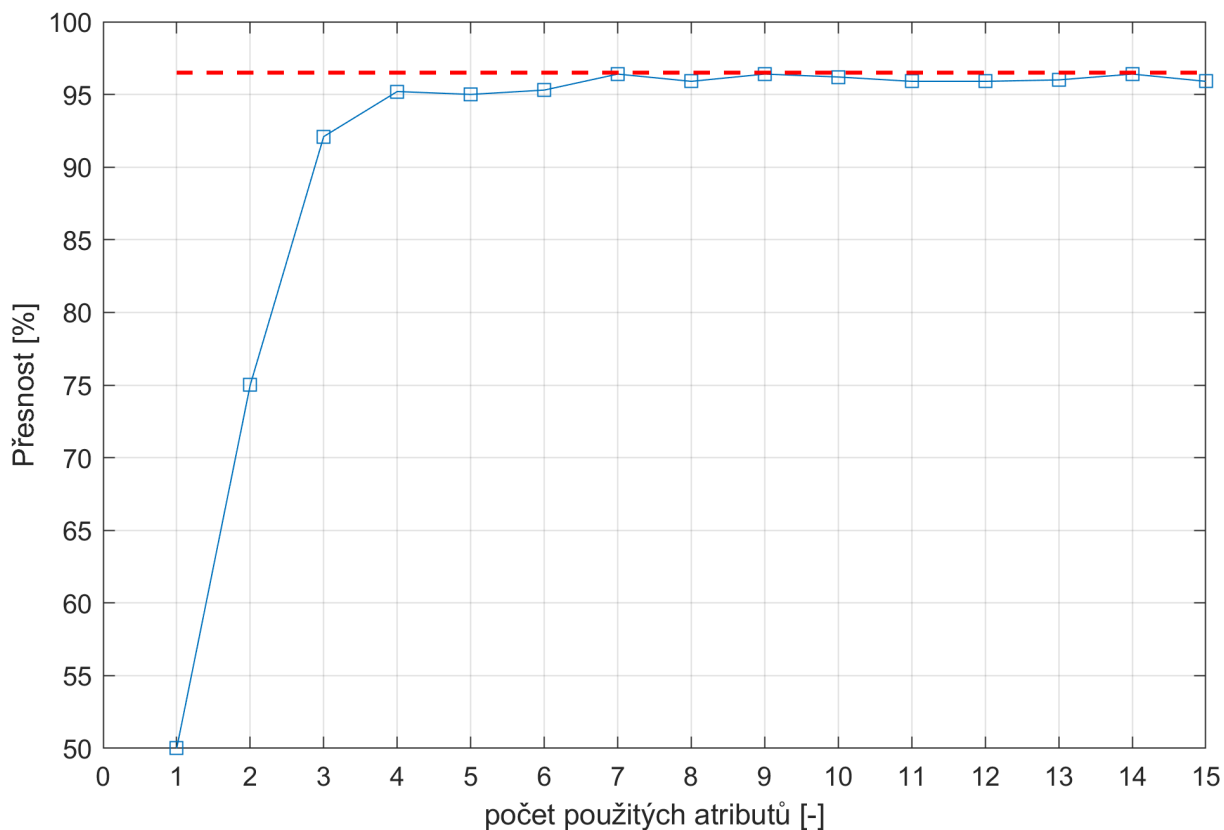
### 3.6.4 SVM volba atributů

Pro přesnější klasifikaci jsem v Classification learneru použil nástroj automatické optimalizace. Ten na sadě 15ti atributů, viz tabulka 3.8 a metodě SVM identifikoval tyto parametry:

Tabulka 3.9: Identifikované hyperparametry.

Multiclass method	One-to-One
Kernel function	linear
Data standaritized	true

Posledním krokem vyhodnocení vhodných atributů bylo testování přesnosti klasifikátoru takovým způsobem, že jsem začal jen s nejlepším atributem a postupně jsem přidával další atributy podle tabulky 3.8. Přesnost takhle trénovaných klasifikátorů pro modelovou situaci 11xxxxx je znázorněna v obrázku 3.7.

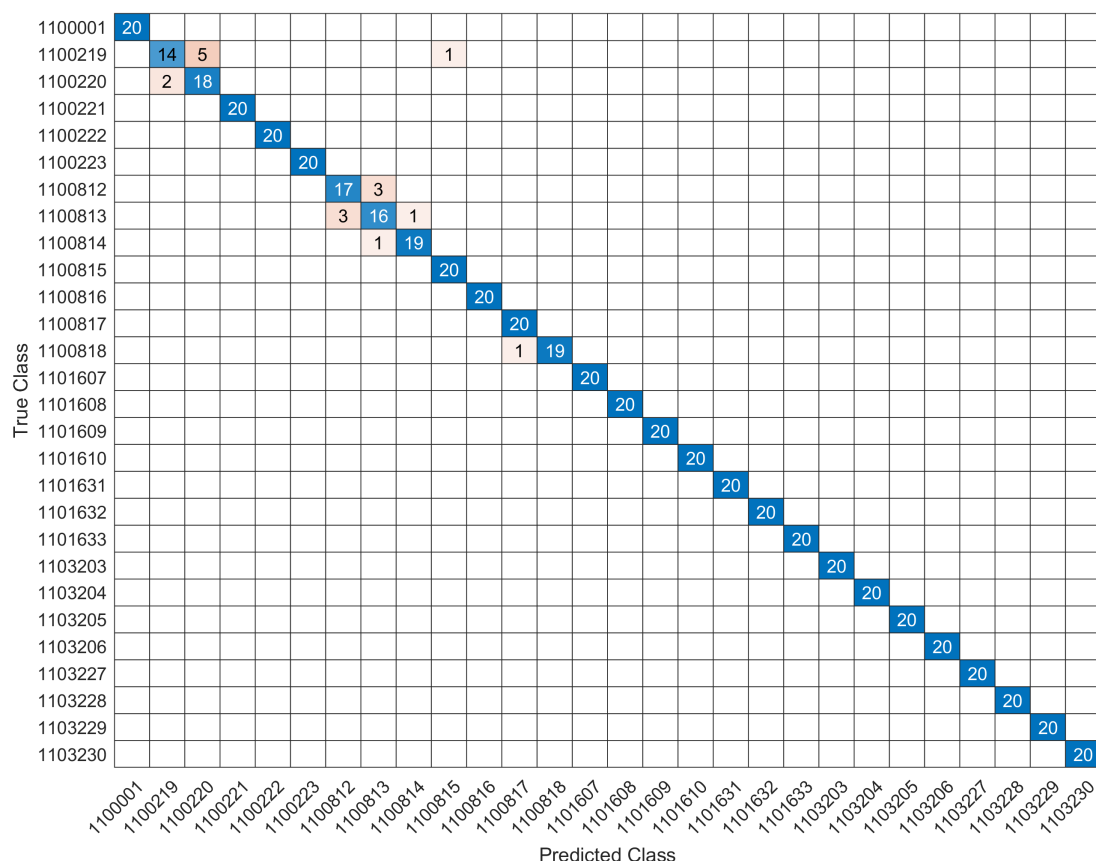


Obrázek 3.7: Závislost přesnosti klasifikátoru na počtu použitých atributů

Z grafu je zřejmé, že přesnost modelu závisí nejvíce na prvních cca 7 attributech tabulka 3.8. Přidávání dalších atributů je zbytečné, protože klasifikátor nedosahuje vyšší přesnosti a z důvodu vyššího počtu atributů jen zbytečně roste čas trénování.

Jako měřítko přesnosti předkládám tabulku misklasifikací pro klasifikátor natrénovaný pro 7 nejlepších atributů na podskupině 11xxxxx. Ta na diagonále ukazuje pro danou třídu (True Class) kolik prvků bylo identifikováno správně, hodnoty mimo diagonálu značí kolikrát a s jakou třídou klasifikátor identifikoval špatně (Predicted Class).

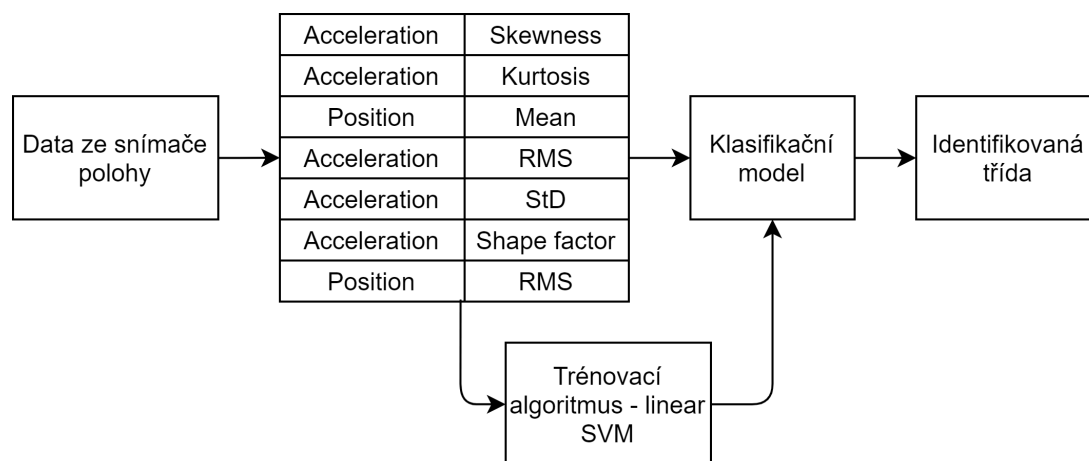
Vidíme, že klasifikátor funguje bezchybně pro omezení průtoku 1 a 2 (třídy 11032xx a 11016xx). Problémy jsou však s identifikací nastavení dolního a horního tlumení. (třídy 11008xx a 11002xx). Myslím si ale, že to není zásadní problém, protože nejčastěji se klasifikátor pletl mezi sousedními třídami, jejichž nastavení je málo rozdílné. Zásadní je, že správně poznal jaký prvek byl nastaven chybně.



Obrázek 3.8: Misklasifikace modelu trénovaného na 7 nejlepších attributech.

### 3.7 Shrnutí algoritmu

Výstupem praktické části je algoritmus, který dokáže identifikovat stavy pneumatického zařízení. Jeho vstupem je signál ze snímače polohy. Z něj je odvozeno zrychlení a následně jsou počítány 7 atributů signálu viz. schéma 3.9. Při aplikaci algoritmu musí proběhnout proces učení na tréninkové sadě dat. Nejvhodnější učicí algoritmus pro tento účel je lineární SVM, jehož parametry sumarizuje tabulka 3.9. Natrénovaný model je schopen identifikovat třídy s přesností pohybující se kolem 96.5%.





Obrázek 3.9: Schéma navrženého algoritmu.

### 3.8 Vytipovaný senzor

Vyhledal jsem dva průmyslově aplikovatelné příklady snímačů polohy pneumatického válce. Oba senzory pracují na principu magnetické indukce, takže je potřeba aby měl válec magnetický píst. Oba umožňují napěťový, proudový nebo IO-link výstup.

Tabulka 3.10: Porovnání dvou vybraných snímačů polohy pneumatického válce. Zdroje: [15],[16].

Výrobce, model	SICK MPA	Balluff BMP
		
Měřicí rozsah (dle modelu)	107-1007 mm	64-256 mm
Rozlišení	0.03%	0.02%
Opakovatelná přesnost	$\pm 0.06\%$	$\pm 0.15\%$
Výstup	0-10V/4-20mA/IO-link	0-10V/4-20mA/IO-link



## 4 Závěr

Cílem této bakalářské práce bylo aplikovat algoritmy prediktivní údržby na data naměřená na experimentálním zařízení za účelem identifikace stavů pneumatického zařízení.

V rešeršní části byla představena koncepce Industry 4.0 a její provázání s prediktivní údržbou. Byl popsán postup implementace prediktivní údržby, blíže byly popsány body týkající se zpracování naměřených dat, výběru vhodných indikátorů stavu a tvorby klasifikačních algoritmů. V poslední části rešerše byl představen Predictive maintenance toolbox a jeho dvě aplikace, které byly použity pro práci s daty jejich vyhodnocování: Diagnostic feature designer a Classification learner.

V úvodu praktické části je popsáno experimentální zařízení, na kterém bylo provedeno měření, způsob a nástroje použité pro sběr dat a modelové situace, které byly simulovány. Po tom, co bylo zařízení zprovozněno a byl rozchozen sběr dat, mohly být provedeny experimenty pokrývající rozsáhlou sadu korektních i chybných stavů.

Hlavní část práce se zabývá vyhodnocováním naměřených dat a identifikací vhodných indikátorů stavu. Data bylo nejprve třeba převést do základních jednotek a následně byly počítány atributy charakterizující data v časové i spektrální oblasti. Z těchto atributů bylo potřeba vybrat ty, které budou spolehlivě identifikovat stavy.

Pro výběr atributů byla použita kombinace filtrace pomocí metody ReliefF a testování na klasifikačním modelu, který byl trénován podle metody SVM. Z výsledků vychází, že pro identifikaci stavu je vhodný snímač polohy. Po tomto zjištění jsem z polohy odvodil rychlost a zrychlení a do procesu tvorby atributů a jejich hodnocení zahrnul i tyto veličiny. To přineslo ještě lepší výsledky pro tento snímač.

Výsledkem zpracování signálů je, že nejvhodnějším senzorem pro identifikaci stavů je lineární enkodér. Jsou stanoveny atributy a metoda a parametry trénování klasifikačního modelu zaručující spolehlivou identifikaci stavu pneumatického zařízení. Nakonec byly vytipovány dva senzory pro průmyslové nasazení.

Další etapou tohoto projektu je nasazení a ověření navržených algoritmů na reálných průmyslových aplikacích, testování a případná optimalizace parametrů.

# Seznam zkratek

**ML** Machine learning

**DA** Diskriminační analýza

**DFD** Diagnostic Feature Designer

**SVM** Support Vector Machine

# Literatura

- [1] Predictive Maintenance with MATLAB. *MathWorks* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.mathworks.com/content/dam/mathworks/ebook/gated/predictive-maintenance-ebook-all-chapters.pdf>
- [2] Overcoming Four Common Obstacles to Predictive Maintenance with MATLAB and Simulink. *MathWorks* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.mathworks.com/content/dam/mathworks/white-paper/gated/predictive-maintenance-challenges-whitepaper.pdf>
- [3] Prediktivní údržba - vytvoření a zavedení modelu do produkce. *Globema* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.globema.cz/prediktivni-udrzba-3/>
- [4] Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowl Inf Syst* **34**, 483–519 (2013). Dostupné z: <https://doi.org/10.1007/s10115-012-0487-8>
- [5] VACULÍK, Karel *Selekce příznaků pomocí nekorelovaných charakteristik*. Brno, 2013. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky.
- [6] Relief (feature selection). *Wikipedia* [online]. [cit. 2021-5-21]. Dostupné z: [en.wikipedia.org/wiki/Relief\\_\(feature\\_selection\)#Relief\\_Algorithm](https://en.wikipedia.org/wiki/Relief_(feature_selection)#Relief_Algorithm)
- [7] Machine Learning: What is Discriminant Analysis? *MATLAB - YouTube* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.youtube.com/watch?v=eBm8Uo9yhwI>
- [8] Support vector machines. *Wikipedia* [cit. 2021-5-21]. Dostupné z: [https://cs.wikipedia.org/wiki/Support\\_vector\\_machines](https://cs.wikipedia.org/wiki/Support_vector_machines)
- [9] Support vector machines (SVM). *MUNI* [online]. [cit. 2021-5-21]. Dostupné z: [https://is.muni.cz/el/1433/podzim2006/PA034/09\\_SVM.pdf](https://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf)
- [10] Support Vector Machines — Soft Margin Formulation and Kernel Trick. *Towards Data Science* [online]. [cit. 2021-5-21]. <https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe>
- [11] Multiclass Classification Using Support Vector Machines. *Baeldung* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.baeldung.com/cs/svm-multiclass-classification>
- [12] Overfitting. *Wikipedia* [online]. [cit. 2021-5-21]. Dostupné z: <https://en.wikipedia.org/wiki/Overfitting>

- [13] Predictive Maintenance Toolbox. *MathWorks* [online]. [cit. 2021-5-21]. [https://www.mathworks.com/help/predmaint/index.html?searchHighlight=predmaint&s\\_tid=srchtitle](https://www.mathworks.com/help/predmaint/index.html?searchHighlight=predmaint&s_tid=srchtitle)
- [14] Druhy údržby staveb. *KVIC* [online]. [cit. 2021-5-21]. Dostupné z: [https://www.kvic.cz/soubor/3345/15-Druhy\\_udrzby\\_staveb.pdf](https://www.kvic.cz/soubor/3345/15-Druhy_udrzby_staveb.pdf)
- [15] Position sensors MPA. *SICK* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.sick.com/ag/en/magnetic-cylinder-sensors/position-sensors/mpa/mpa-107thtp0/p/p305805>
- [16] Magnetic Field Positioning System. *BALLUFF* [online]. [cit. 2021-5-21]. Dostupné z: <https://www.balluff.com/local/us/productfinder/#/ca/A0001/cg/G0105/product/F01508/variant/PV5486461>